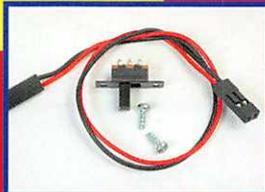
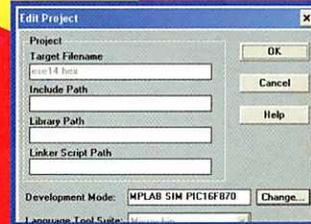
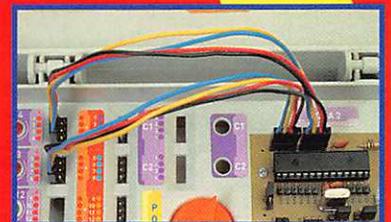
...e costruisci il tuo LABORATORIO DIGITALE

6,90 €

52

HARDWARE

DIGITALE DI BASE

MICROCONTROLLER

DIGITALE AVANZATO

TOTALMENTE PROGRAMMABILE!!!

9 771824 363008 41052

Peruzzo & C.

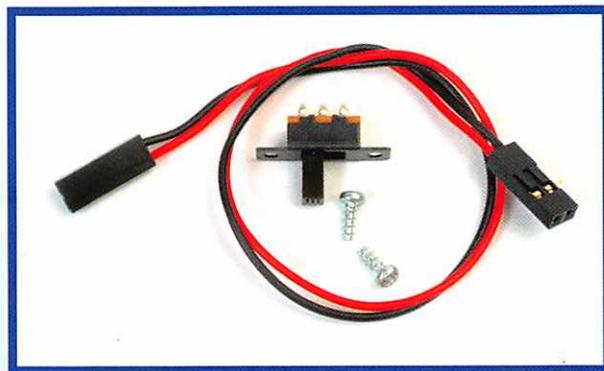
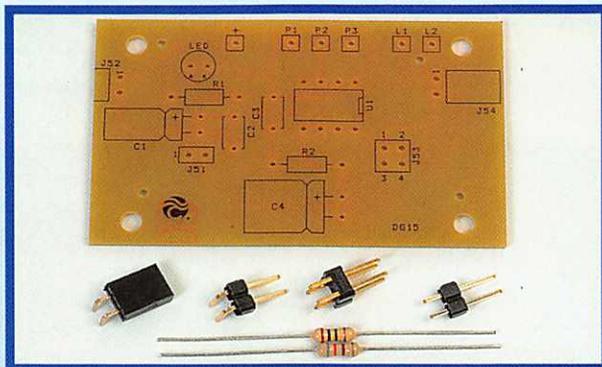
Direttore responsabile:
ALBERTO PERUZZO
Direttore Grandi Opere:
GIORGIO VERCELLINI
Consulenza tecnica e traduzioni:
CONSULCOMP S.n.c.
Pianificazione tecnica
LEONARDO PITTON

Direzione, Redazione, Amministrazione: viale Ercole Marelli 165, Tel. 02/242021, 20099 Sesto San Giovanni (Mi). Pubblicazione settimanale. Registrazione del Tribunale di Monza n. 1738 del 26/05/2004. Spedizione in abbonamento postale gr. II/70; autorizzazione delle Poste di Milano n. 163464 del 13/2/1963. Stampa: Grafiche Porpora s.r.l., Cernusco S/N (MI). Distribuzione SO.DI.P. S.p.A., Cinisello Balsamo (MI).

"ELETTRONICA DIGITALE" si compone di 70 fascicoli settimanali da suddividere in 2 raccoglitori.

RICHIESTA DI NUMERI ARRETRATI. Per ulteriori informazioni, telefonare dal lunedì al venerdì ore 9.30-12.30 all'ufficio arretrati tel. 02/242021. Se vi mancano dei fascicoli o dei raccoglitori per completare l'opera, e non li trovate presso il vostro edicolante, potrete riceverli a domicilio rivolgendovi direttamente alla casa editrice. Basterà compilare e spedire un bollettino di conto corrente postale a PERUZZO & C. s.r.l., Ufficio Arretrati, viale Marelli 165, 20099 Sesto San Giovanni (MI). Il nostro numero di c/c postale è 42980201. L'importo da versare sarà pari al prezzo dei fascicoli o dei raccoglitori richiesti, più le spese di spedizione € 3,10 per pacco. Qualora il numero dei fascicoli o dei raccoglitori sia tale da superare il prezzo globale di € 25,82 e non superiore a € 51,65, l'invio avverrà per pacco assicurato e le spese di spedizione ammonteranno a € 6,20. La spesa sarà di € 9,81 da € 51,65 a € 103,29; di € 12,39 da € 103,29 a € 154,94; di € 14,98 da € 154,94 a € 206,58; di € 16,53 da € 206,58 in su. Attenzione: ai fascicoli arretrati, trascorse dodici settimane dalla loro distribuzione in edicola, viene applicato un sovrapprezzo di € 0,52, che andrà pertanto aggiunto all'importo da pagare. Non vengono effettuate spedizioni contrassegno. Gli arretrati di fascicoli e raccoglitori saranno disponibili per un anno dal completamento dell'opera. IMPORTANTE: è assolutamente necessario specificare sul bollettino di c/c postale, nello spazio riservato alla causale del versamento, il titolo dell'opera nonché il numero dei fascicoli e dei raccoglitori che volete ricevere.

IN REGALO in questo fascicolo

- 1 Commutatore
- 1 Cavetto a 2 fili con due connettori femmina a 2 vie
- 2 Viti

IN REGALO nel prossimo fascicolo

- 1 Scheda DG15
- 1 Resist. 10 Ohm 5% 1/4 W
- 1 Resist. 820 Ohm 5% 1/4 W
- 1 Connettore femmina da c.s. a 90° a 2 vie
- 1 Connettore maschio da c.s. a 90° a 2 vie
- 1 Connettore maschio da c.s. diritto a 2 vie
- 1 Connettore maschio da c.s. diritto a 2 file x 2 vie

COME RACCOGLIERE E SUDDIVIDERE L'OPERA NELLE 4 SEZIONI

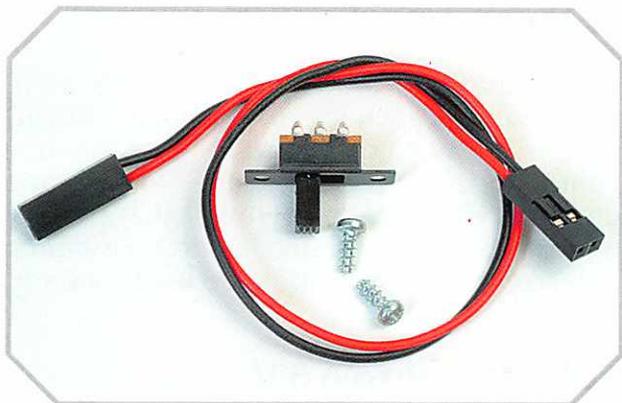
L'Opera è composta da 4 sezioni identificabili dalle fasce colorate, come indicato sotto. Le schede di ciascun fascicolo andranno suddivise nelle sezioni indicate e raccolte nell'apposito raccoglitore, che troverai presto in edicola. Per il momento, ti consigliamo di suddividere le sezioni in altrettante cartellette, in attesa di poterle collocare nel raccoglitore. A prima vista, alcuni numeri di pagina ti potranno sembrare ripetuti o sbagliati. Non è così: ciascuno fa parte di sezioni differenti e rispecchia l'ordine secondo cui raccogliere le schede.
Per eventuali domande di tipo tecnico scrivere al seguente indirizzo e-mail: elettronicadigitale@microrobots.it

Hardware Montaggio e prove del laboratorio

Digitale di base Esercizi con i circuiti digitali

Digitale avanzato Esercizi con i circuiti sequenziali

Microcontroller Esercizi con i microcontroller

Distributore dei 5V (II)

Componenti allegati a questo fascicolo.

Il commutatore si monta in questa zona.

Con questo fascicolo viene fornito il commutatore che manca sul pannello superiore, le due viti per il suo montaggio e un cavetto con due connettori sui terminali.

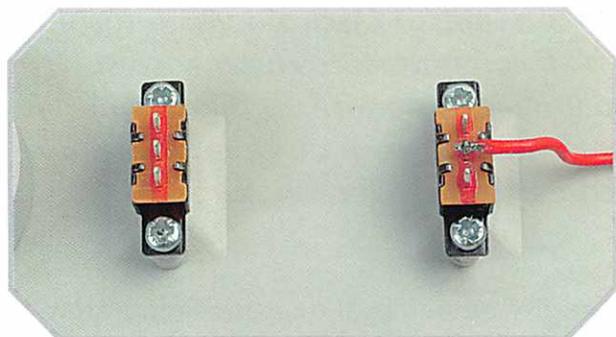
Con il montaggio di questo commutatore si può già portare l'alimentazione da 5 V all'amplificatore audio e si termina la distribuzione dell'alimentazione sul pannello superiore.

Il cavetto

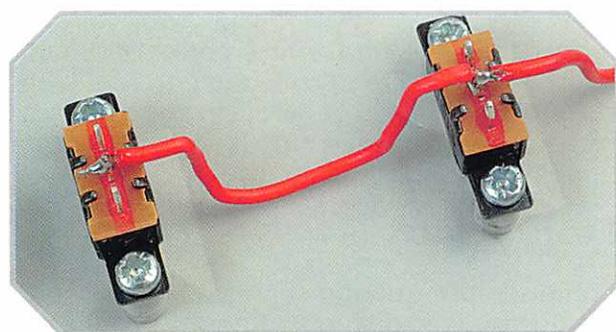
Il cavetto a due fili fornito, permette di ampliare le possibilità di collegamento del laboratorio, è di utilizzo generale e si può utilizzare su connettori maschio sia da due che da quattro vie.

Montaggio

Come sempre quando si lavora all'interno di un dispositivo o di uno strumento, ed è necessario collegare o scollegare parti di circuito o dei pezzi, si deve scollegare completamente l'alimentazione, staccare eventuali alimentatori esterni e le batterie, eliminando così qualsiasi possibilità di errore.

Schema elettrico dell'alimentazione a 5 e 9 V e la sua distribuzione sul pannello superiore.

Commutatore avviato.

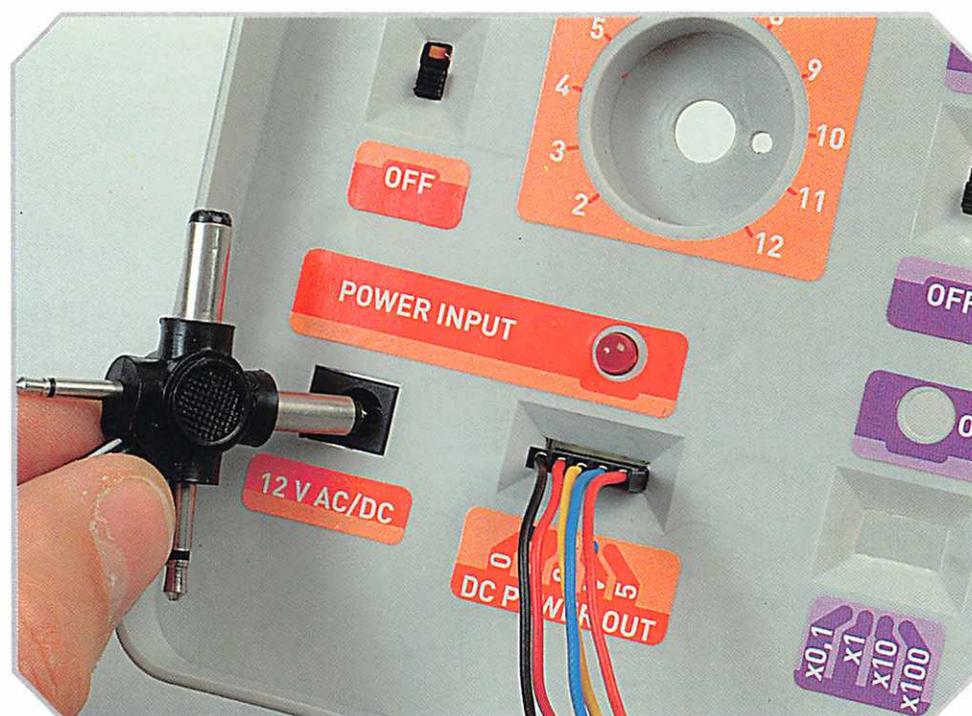
I commutatori

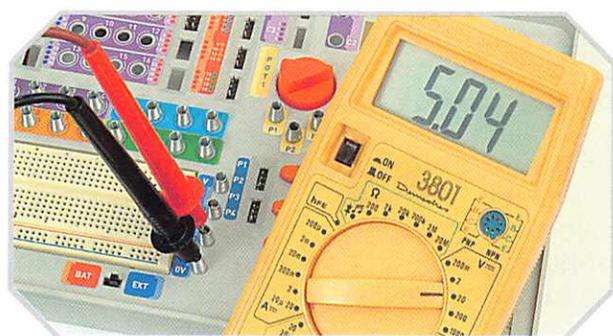
Il commutatore si monta nella posizione corrispondente all'etichetta del pannello AUDIO ON OFF. Se avete già montato i precedenti, è l'unica posizione rimasta libera. Se necessario dovremo eliminare le eventuali sbavature di plastica con l'aiuto di una lametta. Fisseremo poi il commutatore utilizzando le due viti fornite.

Collegamento all'alimentazione da 5 V.

Collegamento a 5 V

Si taglia un pezzo di filo di colore rosso lungo circa 5 cm, lo speleremo da entrambe le parti per 3 o 4 mm, e lo collegheremo fra il terminale centrale di questo commutatore e il terminale centrale di quello che si trova a lato.

Collegamento

Prima di collegare l'alimentatore esterno, dovremo soltanto verificare che il cavetto a cinque fili che unisce i due pannelli del laboratorio, sia collegato correttamente, verificando anche che il colore di ogni filo coincida con quello indicato sui punti dell'etichetta DC POWER OUT del pannello superiore.

Utilizzo di un alimentatore esterno.

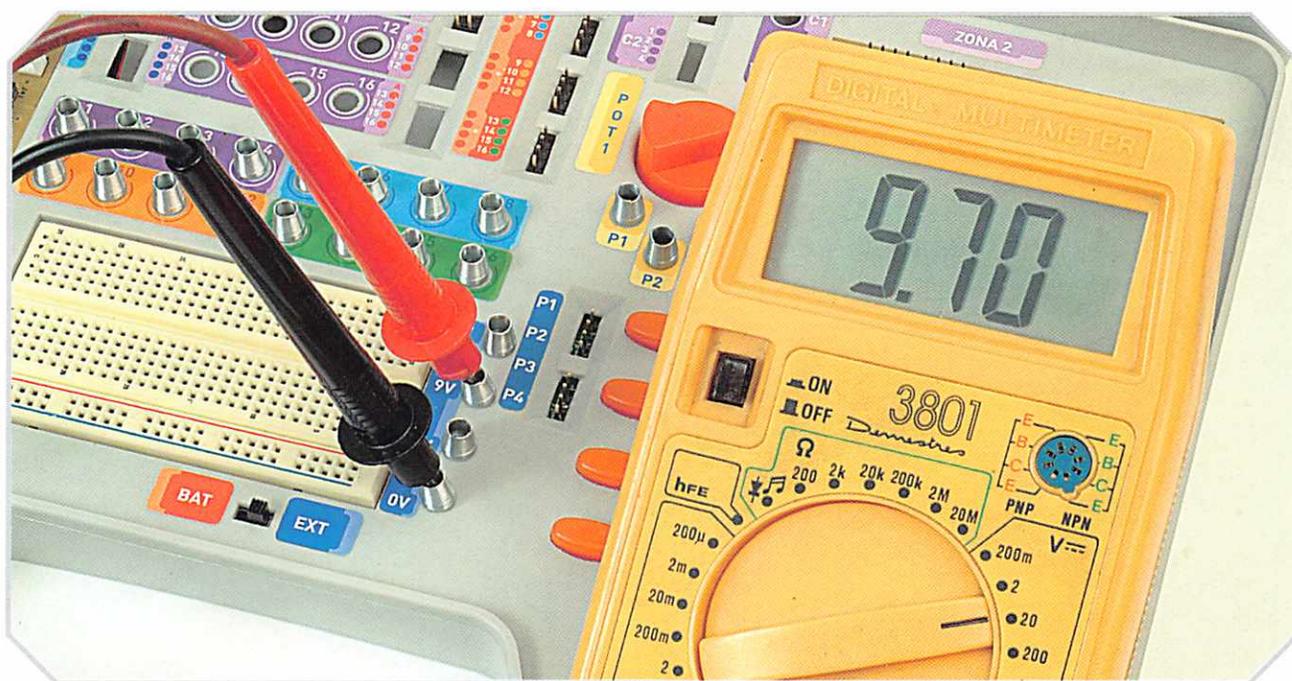
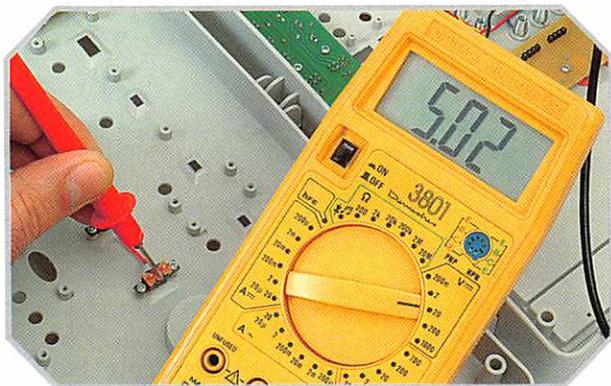
Spostando su ON il commutatore POWER arriva l'alimentazione alla scheda DG18.

Dobbiamo misurare 5 V con il commutatore in EXT.

Alimentazione esterna

Il laboratorio si può alimentare dall'esterno, il tipo di connettore utilizzato è tra i più comuni del suo genere. Vi raccomandiamo di utilizzare un alimentatore che fornisca corrente continua, la polarità in questo caso non è importante, dato che l'ingresso di tensione è dotato di un ponte a diodi seguito da una capacità come filtro. Questa configurazione ci permette di utilizzare anche un trasformatore, e di alimentare con tensione alternata.

La tensione di ingresso a questo alimentatore deve essere quella adatta alla rete di distribuzione disponibile nella vostra località, che in Europa è normalmente 220 V, 50 Hz, tuttavia in altri paesi si utilizza 60 Hz e altre tensioni, ad esempio 110, 125 V ecc. Questo alimentatore non viene fornito, ogni lettore ne dovrà acquistare uno, o utilizzarne uno recuperato, tenendo presente la tensione e la frequenza del posto in cui ci si trova.

La tensione di uscita dell'alimentatore, che deve essere collegata all'ingresso del laboratorio è di 12 V, alternata oppure continua.

Per realizzare l'alimentazione esterna il commutatore EXT/BAT deve essere sulla posizione EXT. Se è in posizione BAT si alimen-

Alimentazione da 9/10 V.

Verifica della linea di alimentazione da 5 V dei commutatori.

Sulla posizione BAT si alimenta solo tramite batterie.

ta tramite le batterie anche se è presente l'alimentazione esterna.

Altre opzioni

Nel caso in cui non si disponga di questo tipo di trasformatore, o se ne voglia utilizzare un altro, ad esempio da 8 V, lo si può fare, ma in questo caso si dovrà utilizzare solamente la tensione da 5 V, tenendo presente che con questa tensione è possibile utilizzare tutto il laboratorio. Come regola generale è necessario che la tensione sul terminale di ingresso di ogni regolatore, sia almeno 2 Volt superiore a quella di uscita. La tensione misurata sul punto 9 V può arrivare a 10 V, e fa parte di quelle utilizzate dai costruttori CMOS per fornire i dati tecnici degli stessi. Gli esperimenti proposti sul laboratorio funzionano perfettamente sia a 9 che a 10 Volt, ma se si desidera regolare alla tensione di 9 V, è possibile abbassare il valore della resistenza R2 della scheda DG18, oppure aumentarla per fare alzare la tensione sul terminale 9 V.

Consumo

Le schede DG18 e DG19 sono progettate per un consumo massimo di 100 mA per ogni regola, cioè per 5 V e per 9 V.

Più avanti per i più esperti spiegheremo come si può aumentare la corrente su queste uscite.

Vista generale del laboratorio.

Astabile con tre porte NOR

Le porte NOR possono anche essere usate nei circuiti oscillatori astabili, esistono molti schemi possibili, anche se non tutti hanno un funzionamento corretto. Quello proposto fa parte di quelli che funzionano meglio.

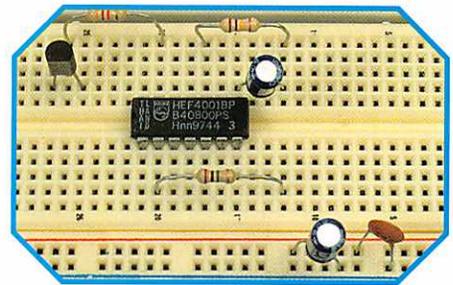
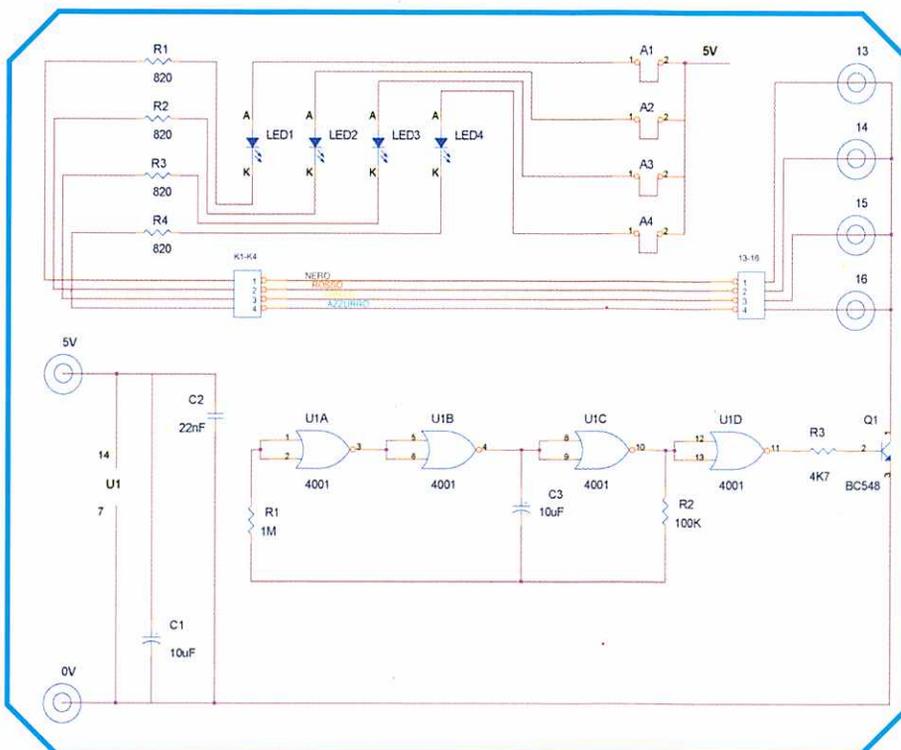
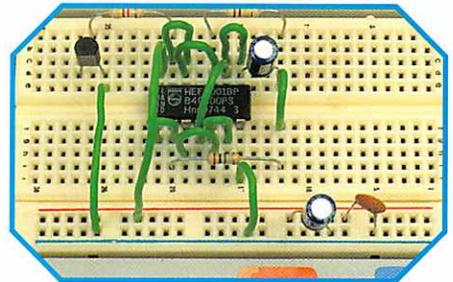
Il circuito

L'astabile è formato dalle porte U1A, U1B e U1C del circuito integrato 4001. La frequenza dipende dal valore della resistenza R2 e dalla capacità del condensatore C3. Bisogna tenere presente anche che la resistenza R1 deve essere maggiore di R2. La frequenza approssimata si ottiene con la formula seguente:

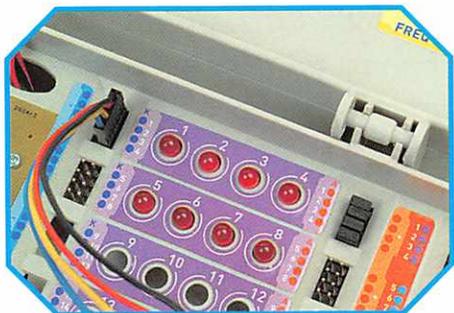
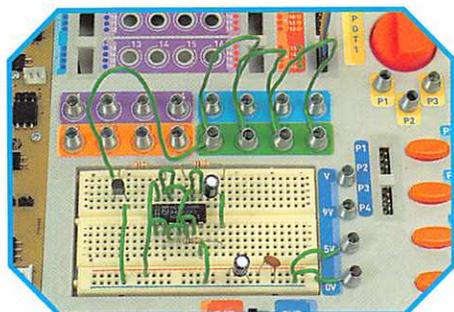
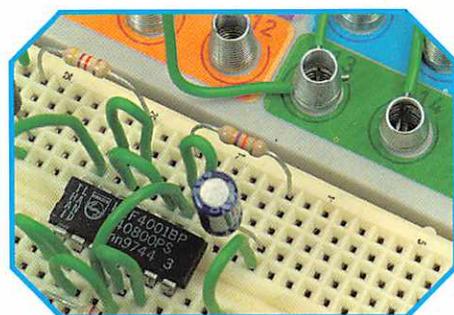
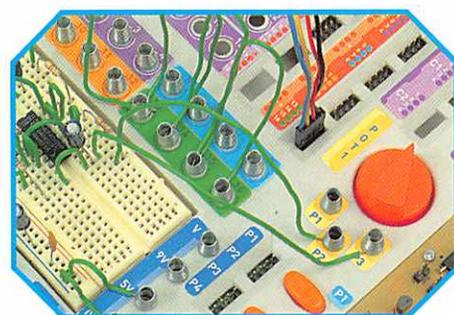
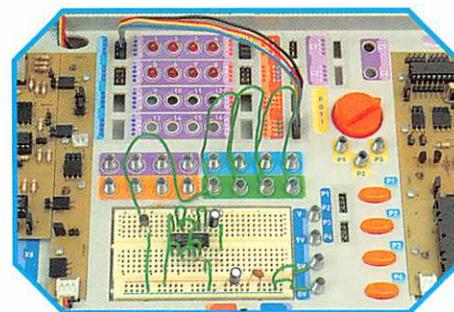
$$f = 2{,}2/R2C$$

La resistenza va espressa in Ohm e la capacitò in Farad per poter ottenere il valore di frequenza in Herz.

Se R2 è da 100 K e C3 da 10 µF si ottiene una frequenza di 2,2 Hz, sufficientemente bassa per poter verificare il funzionamento dell'oscillatore, dato che sarà possibile osservare come si spegne o si accende un diodo LED. Se ad esempio utilizzassimo rispettivamente 47 K e

Componenti sulla scheda Bread Board.

Collegamenti della scheda Bread Board.

LISTA DEI COMPONENTI

U1	Circuito integrato 4001
R1	Resistenza 1 M (marrone, nero, verde)
R2	Resistenza 100 K (marrone, nero, giallo)
R3	Resistenza 4K7 (giallo, viola, rosso)
R2	Resistenza 47 K (giallo, viola, arancio)
C1, C3	Condensatore 10 µF elettrolitico
C2	Condensatore 22 nF

Schema del circuito astabile.

È necessario collegare i ponticelli degli anodi.

Collegamento fra la scheda Bread Board e le molle.

Cambiando R2 cambia la frequenza.

Variazione di frequenza con il potenziometro.

Esperimento completato.

100 nF, si otterrebbe una frequenza udibile di circa 400 Hz, cosa che potremo verificare quando avremo a disposizione l'amplificatore audio e l'altoparlante. Continuando con lo schema possiamo vedere che l'uscita dell'oscillatore si collega a una quarta porta U1D, che svolge la funzione di stadio di separazione, per evitare che il consumo del circuito possa influenzare il funzionamento dell'oscillatore. In questo modo si ottiene che il carico dell'oscillatore sia rappresentato da una porta CMOS ad alta impedenza di ingresso. Nel caso volessimo utilizzare un carico elevato, ad esempio quattro LED, sicuramente eccessivo per una porta, potremo utilizzare un transistor il quale può condurre diversi milliampère. La resistenza di base R3, è sufficientemente bassa per assicurare che il transistor si saturi, fatto indispensabile per ottenere una buona illuminazione del LED, evitando nel contempo il riscaldamento del transistor stesso, dato che la tensione di saturazione collettore/emettitore è bassa.

Montaggio

Il montaggio è abbastanza veloce, bisogna stare attenti però agli eccessi di confidenza e seguire attentamente gli schemi, facendo attenzione alla polarità dei condensatori elettrolitici, senza dimenticare di collegare l'alimentazione del circuito integrato e inserire i ponticelli sugli anodi dei LED. Questo circuito si alimenta a 5 V.

L'esperimento

Il circuito deve iniziare a funzionare collegando l'alimentazione, in quel momento i LED devono iniziare a lampeggiare. Possiamo fare delle prove cambiando la frequenza, a questo scopo possiamo sostituire la resistenza R2 con una da 47 K, oppure è possibile inserire in serie a questa resistenza il potenziometro P1, per poter variare la frequenza stessa tramite la manopola del potenziometro. È necessario tener presente che se la frequenza sale, non sarà più possibile seguire a vista l'intermittenza del LED perché lo percepiremo sempre illuminato. Questo circuito permette di ottenere frequenze in tutta la banda audio, ma è necessario attendere di avere a disposizione l'amplificatore audio e l'altoparlante per poterne verificare il funzionamento.

Esercizio 13: convertitore A/D, la pratica

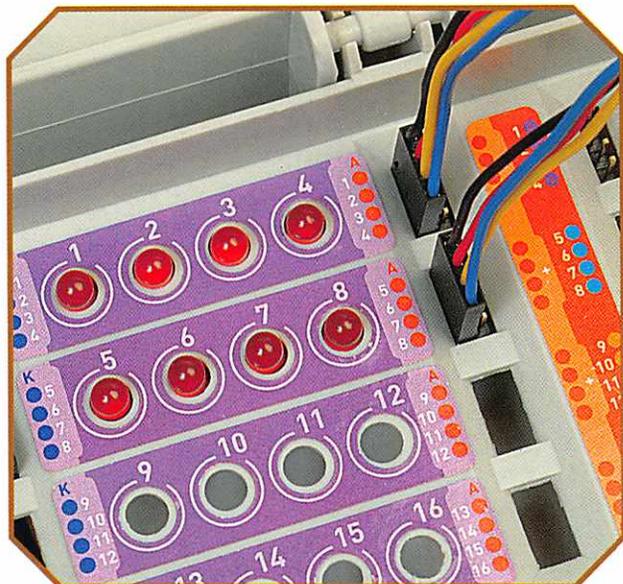
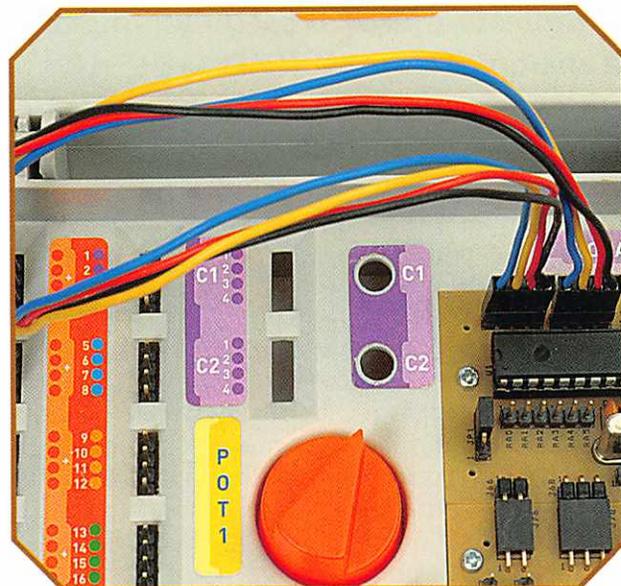
***In** questo esercizio eseguiremo il montaggio di un circuito invertitore analogico/digitale (CAD, ADC, A/D). Nel fascicolo precedente abbiamo realizzato un programma che convertiva un segnale analogico in uno digitale e lo rappresentava in binario mediante la matrice dei LED. Eseguiremo il montaggio di questa applicazione e ne proveremo il funzionamento.*

Scrittura del microcontroller

Quando compiliamo il codice con MPLAB otteniamo il file nel codice macchina corrispondente all'esercizio 13 (ese13.hex), ora dobbiamo scrivere questo file sul microcontroller. La prima cosa da fare sarà configurare l'hardware del nostro laboratorio. Verificheremo che il PIC sia inserito correttamente nel suo zoccolo, la presenza dei ponticelli JP8 e JP9 e che i connettori JP1, JP2 e JP3 siano sulle posizioni 1 e 2. Collegheremo il laboratorio e il PC tramite il cavo di comunicazione, e faremo partire il software IC-Prog, a questo punto siamo pronti per eseguire il trasferimento.

Ricordiamo i passaggi che bisogna seguire per scrivere un programma sul microcontroller: lettura, cancellazione, verifica, scrittura e verifica. Dopo aver eseguito le prime tre fasi, apriremo il file che vogliamo scrivere sul PIC e

```
IC-Prog 1.05C - Programmatore Prototipo - C:\Programmi\MPLAB\Progetti\ESE13.HEX
File  Modifica  Buffer  Settaggi  Comandi  Strumenti  Visualizza  Aiuto
                                                   PIC 16F870
Address - Program Code
0000: 2808 3FFF 3FFF 3FFF 3FFF 081E 0086 0008
0008: 0185 0186 3051 009F 1683 3000 009F 0186
0010: 30CF 0081 30FF 0085 1283 0064 0186 130C
0018: 151F 191F 2819 2005 2815 3FFF 3FFF 3FFF
0020: 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF
0028: 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF
0030: 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF
0038: 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF
0040: 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF
0048: 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF
0050: 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF
0058: 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF
Address - Eeprom Data
0000: FF FF FF FF FF FF FF FF
0008: FF FF FF FF FF FF FF FF
0010: FF FF FF FF FF FF FF FF
0018: FF FF FF FF FF FF FF FF
0020: FF FF FF FF FF FF FF FF
0028: FF FF FF FF FF FF FF FF
0030: FF FF FF FF FF FF FF FF
0038: FF FF FF FF FF FF FF FF
Configurazione
Oscillator: XT
Code Protect: CP OFF
Fuses: [x] WDT  [x] PWRT  [ ] BODEN  [ ] LVP  [ ] CPD  [ ] WRT  [ ] DEBUGGER
Checksum: C763   ID Value: FFFF
Config Word : 3D35h
Buffer 1 | Buffer 2 | Buffer 3 | Buffer 4 | Buffer 5
JDM Programmer su Com1   Chip: PIC 16F870 (E9)
```

Apriamo il file e configuriamo IC-Prog.

Colleghiamo i cavetti alla matrice dei LED.

Colleghiamo la matrice dei LED alle uscite del PIC.

configureremo l'oscillatore, la protezione del codice e i bit della parola di configurazione. Ora IC-Prog avrà l'aspetto della figura riportata nella pagina precedente.

Selezioneremo "Programma Tutto" e il programma si trasferirà al PIC. In ultimo, verificheremo che la scrittura sia stata eseguita correttamente leggendo il dispositivo.

Leggiamo il PIC per verificare la fase di scrittura.

Montaggio

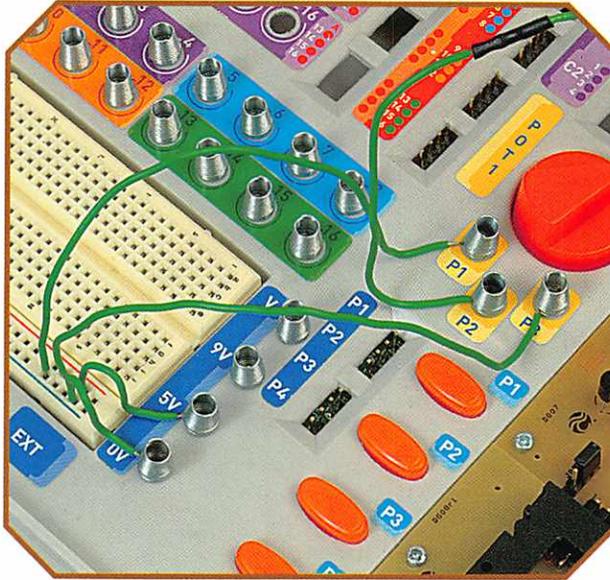
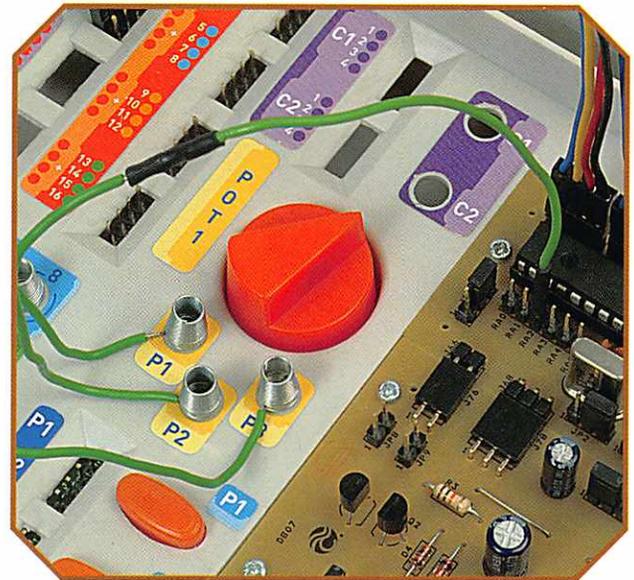
Configurazione del laboratorio

Dobbiamo preparare il laboratorio per provare il funzionamento dell'applicazione, per questo ripristineremo la configurazione precedente a quella richiesta per la scrittura. Scollegheremo il cavo di trasferimento, toglieremo i ponticelli dai connettori JP8 e JP9 e sposteremo sulle posizioni 2 e 3 i ponticelli dei connettori JP1, JP2 e JP3.

Le uscite

L'applicazione mostrerà, mediante la matrice dei LED, il risultato della conversione di un se-

OPZIONE 1				OPZIONE 2			
RB0	RB1	RB2	RB3	RB3	RB2	RB1	RB0
○	○	○	○	○	○	○	○
RB4	RB5	RB6	RB7	RB7	RB6	RB5	RB4
○	○	○	○	○	○	○	○

Due possibilità di visualizzare le uscite.

Colleghiamo il potenziometro all'alimentazione.

Montaggio completo dell'ingresso analogico.

gnale analogico nel suo corrispondente digitale. Questo risultato verrà fornito in formato binario a 8 bit, avremo quindi bisogno di due file di LED. Dobbiamo inserire i ponticelli sui catodi, sulla parte sinistra della matrice, e collegare i due cavetti a quattro fili agli anodi. Possiamo vedere questo collegamento nelle immagini della pagina precedente.

La porta del microcontroller che utilizzeremo come uscita è la porta B. La parte libera dei due cavetti verrà collegata a questa porta. Faremo corrispondere la fila superiore della matrice con i quattro bit meno significativi della porta (RB3:RB0) e la fila inferiore con i quattro più significativi (RB7:RB4). Questo collegamento può essere fatto in due modi, come possiamo vedere anche dalla figura, dove nella prima opzione il diodo 1 corrisponde a RB0, il 2 a RB1 e così via; oppure possiamo incrociare i cavi in modo che il diodo 4 corrisponda con RB0, il 3 con RB1 ecc.. Quest'ultima rappresentazione dell'uscita sarebbe la più adatta a interpretare il risultato, in quanto lavorando con un formato binario i bit meno significativi rimarrebbero a destra.

Schema elettrico di un potenziometro.

L'ingresso

Il segnale di ingresso dell'applicazione è un segnale analogico che otteniamo dal potenziometro. Un potenziometro è una resistenza variabile, che risponde allo schema elettrico della figura. A uno dei capi della resistenza collegheremo l'alimentazione (5 Volt) e all'altro capo la massa o 0 V. In funzione della posizione del cursore otterremo un valore di tensione su quest'ultimo, dato che funziona come un partitore di tensione.

Dalla scheda Bread Board, con un filo collegheremo la molla di collegamento a 5 V e con un altro filo a quella dello 0 V. Con due fili più lunghi porteremo i due segnali ai capi del potenziometro. Nella figura possiamo vedere il montaggio descritto, dove i 5 V sono collegati

Muoviamo il potenziometro per cambiare il valore dell'ingresso.

Montaggio finale dell'applicazione.

alla molla P3 e 0 V alla molla P1. Dalla molla di collegamento P2, che corrisponde al cursore del potenziometro, otterremo il segnale analogico che dobbiamo collegare all'ingresso.

Attraverso un altro filo di collegamento uniremo P2 con il pin della porta A, configurato come ingresso del convertitore (RA2). L'ingresso sarà correttamente collegato dopo aver terminato il montaggio appena descritto.

Prova del funzionamento

Per provare il corretto funzionamento del convertitore, è sufficiente ruotare il potenziometro e osservare il risultato sulla matrice dei LED. Con il potenziometro completamente ruotato da una parte, otterremo un ingresso di 5 volt, quindi il risultato della conversione sarà '11111111' il che significa avere tutti i LED accesi. Ruotandolo dall'altra parte applicheremo all'ingresso 0 V, quindi il risultato della conversione sarà '00000000', di conseguenza tutti i LED saranno spenti. Il PIC16F870 dispone di un convertitore a 10 bit, con i quali è possibile rappresentare 1.024 valori (2^{10} = 1.024). Se l'ingresso è compreso tra 0 e 5 V, la precisione del convertitore è di 5/1.024 = 4,88 mV.

Questo significa che ogni variazione di 4,88 mV sull'ingresso si rifletterà sul risultato della conversione.

Dato che nel nostro caso stiamo visualizzando solamente gli 8 bit più significativi del risultato della conversione, la precisione sarà di 5V/2^8 = 5 V/256 = 19,5 mV, quindi ogni 19,5 mV potremo visualizzare una variazione sull'uscita.

Verificheremo inoltre che sono sufficienti piccole variazioni del potenziometro per generare cambiamenti sull'uscita.

Applicazioni

Immaginate di utilizzare al posto del potenziometro una PT100. La PT100 è una resistenza variabile in funzione della temperatura. Supponete che il range di lavoro di questa resistenza sia fra –20° e 100°, quindi il nostro convertitore avrà una precisione di 120°/1.024 = 0,11°. Rileveremo ogni variazione di 0,11°, questo ci permetterà di visualizzare la temperatura con la precisione di un decimo di grado (sul display o su un LCD), controllare il processo in cui è inserita la resistenza con un controllo molto sensibile e preciso.

Immaginate ora di voler costruire invece di un termometro un sensore di pressione digitale o un sensore di umidità. Qualsiasi segnale di tipo analogico si può convertire in uno digitale ed elaborarlo, questo, grazie alle CAD del PIC16F870, si può fare in un modo molto semplice.

Esercizio 14: la memoria EEPROM, il programma

Nell'esercizio che vi presentiamo si lavora con la memoria EEPROM. Verrà sviluppato un programma che ci servirà per imparare a lavorare con questa memoria e poterla così utilizzare in applicazioni più complesse.

La memoria EEPROM

La memoria dei dati è formata da una memoria RAM e una memoria EEPROM. Nella RAM si scrivono i dati di utilizzo generale, che sono gestiti nel programma e i registri specifici i cui bit controlleranno il funzionamento del processore e i dispositivi interni. La memoria EEPROM (non volatile) si utilizza per scrivere i dati i cui valori devono essere mantenuti anche quando si toglie l'alimentazione. Questa memoria è molto utilizzata per applicazioni in cui è richiesta una chiave di sicurezza, come ad esempio un controllo di accessi.

Enunciato

Si tratta di sviluppare un programma che ottenga il valore dello stato degli interruttori RA4-RA0, lo scriva nella EEPROM quando RA4 è attivo e visualizzi il contenuto della EEPROM sulla matrice dei LED che sarà collegata alla porta B.

Come si può vedere l'esercizio previsto è un semplice programma, la cui unica difficoltà consiste nella gestione della memoria EEPROM.

Organigramma

Per sviluppare il codice che risolve l'applicazione, dobbiamo tenere presente che lavoreremo con ingressi e uscite digitali, e che utilizzeremo la EEPROM, in modo da poter scrivere su di essa e poterne leggere il contenuto. Per questa ragione separeremo il codice in tre parti differenti:

- Programma principale: configureremo i dispositivi comuni, che in questo caso saranno le porte di ingresso e di uscita.
- Scrittura: programmeremo la scrittura nella memoria EEPROM.
- Lettura: otterremo il valore desiderato di questa memoria e agiremo sulle uscite.

Nella figura in basso, si può vedere l'organigramma che corrisponde al proponimento appena esposto.

Codice

Programma principale

Cominceremo a programmare con i commenti dell'intestazione del programma, nel quale spiegheremo la funzionalità di quest'ultimo. Proseguiremo con la definizione del PIC, inclu-

IL PIC16F870 ha tre tipi di memoria.

Organigramma dell'applicazione.

```
;                        ESERCIZIO: EEPROM
;                     ------------------------
;
;Programma che rileva il valore dello stato degli interruttori
;RA4-RA0, lo scrive nella EEPROM quando si attiva RA4 e visualizza
;il contenuto della EEPROM sulla barra dei LED della porta B.

        LIST     P=16F870          ;Definiamo il nostro PIC
        INCLUDE  "P16F870.INC"     ;File dei registri interni

        ORG      0
        GOTO     INIZIO
        ORG      5

INIZIO  clrf     PORTB
        bsf      STATUS,RP0        ;Passiamo al banco 1
        movlw    b'00000110'
        movwf    ADCON1            ;Configuriamo la Porta A come I/O digitali
        clrf     TRISB             ;Porta B uscita
        movlw    b'00011111'
        movwf    TRISA             ;RA0-RA4 ingressi
        bcf      STATUS,RP0        ;Passiamo nuovamente al banco 0

CICLO   btfsc    PORTA,4           ;Se l'interruttore è attivo memorizziamo
        call     SCRITTURA         ;Salta alla routine di scrittura
        call     LETTURA           ;Salta alla routine di lettura
        goto     CICLO             ;Ciclo infinito
```

Codice del programma principale.

```
;il contenuto della EEPROM sulla barra dei LED della porta B.

        LIST     P=16F870          ;Definiamo il nostro PIC
        INCLUDE  "P16F870.INC"     ;File dei registri interni

EE_Dir  equu     0x20
EE_Dato equu     0x21

        ORG      0
        GOTO     INIZIO
        ORG      5
```

Definiamo le due variabili all'inizio del codice.

dendo la libreria che contiene la definizione dei registri. Mediante le direttive ORG organizzeremo la memoria di programma e a partire dall'etichetta Inizio, inizieremo a configurare i dispositivi.

Useremo la porta B come uscita, elimineremo quindi qualsiasi valore residuo che ci potrebbe essere mediante l'istruzione clrf PORTB, e tramite il registro TRISB indicheremo che i suoi pin saranno utilizzati come uscite.

La porta A lavorerà come ingresso, quindi configureremo il registro TRISA e definiremo il tipo di ingresso mediante il registro ADCON1. Creeremo un ciclo in cui in funzione del valore del pin RA4, si esegue la chiamata alla subroutine di scrittura e a quella di lettura, o solamente a quella di lettura.

Il programma principale terminerà con questo codice (vedi immagine in alto), ripetendo continuamente il ciclo finale.

Subroutine di scrittura nella EEPROM

Quando RA4 è a 1, livello alto, dobbiamo passare il valore presente sull'ingresso alla memoria EEPROM. Creeremo a questo scopo due variabili, una che contiene il dato da scrivere e l'altra per l'indirizzo dove verrà scritto. Il contenuto di queste due variabili si deve passare ai corrispondenti registri della EEPROM, EEDATA e EEADR. Non bisogna dimenticare che ogni volta che si completa una scrittura su questa memoria, un bit di flag si attiva, per indicarci

```
SCRITTURA
        movf    PORTA,W
        movwf   EE_Dato         ;Acquisiamo lo stato di RA4-0
                                ;e lo passiamo a EE_Dato (registro di appoggio)
        bcf     PIR2,EEIF       ;Resettiamo il flag della EEPROM
        bsf     STATUS,RP1      ;Passiamo al banco 2
        bcf     STATUS,RP0
        movlw   EE_Dir
        movwf   EEADR           ;Indirizzo della EEPROM in cui scrivere
        movf    EE_Dato,W
        movwf   EEDATA
        bsf     STATUS,RP0      ;Passiamo al banco 3
        bcf     EECON1,EEPGD    ;Selezioniamo la EEPROM dei dati
        bsf     EECON1,WREN     ;Abilitiamo la scrittura della EEPROM
        movlw   0x55
        movwf   EECON2
        movlw   0xaa
        movwf   EECON2          ;Sequenza obbligatoria
        bsf     EECON1,WR       ;Iniziamo la scrittura
        bcf     STATUS,RP0
        bcf     STATUS,RP1      ;Torniamo al banco 0
ATTENDI  btfss  PIR2,EEIF       ;Attendiamo il termine della scrittura
         goto   ATTENDI
         return
```

Codice della subroutine di scrittura nella EEPROM.

```
LETTURA bsf     STATUS,RP1      ;Passiamo al banco 2
        bcf     STATUS,RP0
        movlw   EE_Dir
        movwf   EEADR           ;Scrive l'indirizzo
        bsf     STATUS,RP0      ;Passiamo al banco 3
        bcf     EECON1,EEPGD    ;Selezioniamo la EEPROM dei dati
        bsf     EECON1,RD       ;Abilitiamo la lettura della EEPROM
        bcf     STATUS,RP0      ;Torniamo al banco 2
        movf    EEDATA,W        ;Leggiamo il dato
        bcf     STATUS,RP1      ;Torniamo al banco 0
        movwf   PORTB           ;Visualizziamo il dato letto sui LED
        return
```

Codice della subroutine di lettura.

che il processo è terminato. Questo bit deve essere resettato cioè impostato a zero, ogni volta che entriamo in questa subroutine.

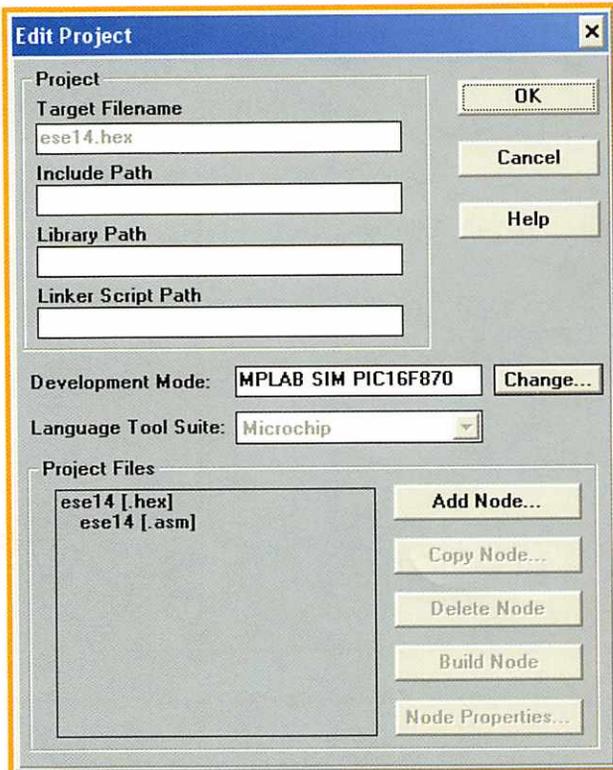
Configureremo la memoria mediante il registro EECON1, per selezionare la EEPROM dei dati e abilitare la scrittura in essa, e in seguito configureremo il registro EECON2 con la sequenza di istruzioni da utilizzare per lavorare con questa memoria in modo scrittura. Questa sequenza sarà sempre la stessa e dovrà essere utilizzata sempre in questo modo.

Siamo pronti per scrivere, inizieremo quindi impostando a 1 il bit WR del registro EECON1. Dato che questa operazione impiega un certo tempo, resteremo all'interno di un ciclo in attesa che il flag che indica il termine di questo processo passi a 1.

Subroutine di lettura

Capita la subroutine precedente, che comprende la lettura del dato nella EEPROM e le operazioni sull'uscita, risulta molto più semplice. Per leggere un dato della memoria la prima cosa da sapere è la posizione dove si trova questo dato. Sul registro EEADR caricheremo l'indirizzo che si trova nella variabile creata espressamente per questo scopo. Configureremo la memoria come nella subroutine precedente, ma con la differenza che ora vogliamo leggere in essa, dopo attiveremo il bit RD del registro EECON1. Il dato si troverà sul registro EEDATA, quindi lo sposteremo sul registro di lavoro W.

Fatto questo abbiamo già a disposizione il

```
Edit Project
Project
Target Filename: ese14.hex
Include Path:
Library Path:
Linker Script Path:
Development Mode: MPLAB SIM PIC16F870   Change...
Language Tool Suite: Microchip
Project Files
ese14 [.hex]
  ese14 [.asm]
OK  Cancel  Help
Add Node...  Copy Node...  Delete Node  Build Node  Node Properties...
```

Raccomandiamo di dare al progetto lo stesso nome del codice in assembler.

dato, ora lo possiamo utilizzare come desideriamo. Nel nostro caso lo passeremo semplicemente sulla porta di uscita, la porta B.

Il codice non è ancora terminato, infatti bisogna ancora indicare il punto in cui deve finire. Inseriamo la direttiva END al termine del codice per poterlo dichiarare terminato.

Compilazione

Apriamo MPLAB e inseriamo il nostro codice in un progetto. È sempre consigliabile che il nome del progetto coincida con quello del codice, in questo modo eviteremo delle confusioni. Compiliamo l'insieme per poter ottenere il file in codice macchina necessario per scrivere il PIC. Se il programma non ha errori verrà creato il file ".hex" desiderato, nel caso ce ne fossero sarà necessario mettere a punto il programma risolvendoli uno per uno. Come potete vedere nella figura in basso, il codice si assembla e si compila con successo. Il compilatore ci informa, tramite dei messaggi, che i banchi di memoria in cui abbiamo localizzato parte dei registri, non concordano con quelli di default. Questo perché il compilatore utilizza la struttura di memoria del PIC 16F84, e non coincide con quella del nostro (il PIC 16F84 ha solamente due banchi di memoria).

Questi messaggi sono unicamente a scopo informativo e non impediscono che la compilazione avvenga con successo, ma devono essere presi in considerazione se per qualche motivo, legato ad altri aspetti della programmazione, dovessimo fare riferimento alla localizzazione dei registri sui banchi di memoria.

```
Build Results
Building ESE14.HEX...

Compiling ESE14.ASM:
Command line: "C:\PROGRA~1\MPLAB\MPASMWIN.EXE /p16F870 /q C:\PROGRA~1\MPLAB\PROGETTI\ESE14.ASM"
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\ESE14.ASM 22 : Register in operand not in bank 0.  Ensure that ba
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\ESE14.ASM 23 : Register in operand not in bank 0.  Ensure that ba
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\ESE14.ASM 25 : Register in operand not in bank 0.  Ensure that ba
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\ESE14.ASM 41 : Register in operand not in bank 0.  Ensure that ba
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\ESE14.ASM 43 : Register in operand not in bank 0.  Ensure that ba
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\ESE14.ASM 45 : Register in operand not in bank 0.  Ensure that ba
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\ESE14.ASM 46 : Register in operand not in bank 0.  Ensure that ba
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\ESE14.ASM 48 : Register in operand not in bank 0.  Ensure that ba
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\ESE14.ASM 50 : Register in operand not in bank 0.  Ensure that ba
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\ESE14.ASM 51 : Register in operand not in bank 0.  Ensure that ba
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\ESE14.ASM 62 : Register in operand not in bank 0.  Ensure that ba
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\ESE14.ASM 64 : Register in operand not in bank 0.  Ensure that ba
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\ESE14.ASM 65 : Register in operand not in bank 0.  Ensure that ba
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\ESE14.ASM 67 : Register in operand not in bank 0.  Ensure that ba

Build completed successfully.
```

Risultato della compilazione.